# 物联网智能水阀微信小程序开发文档

## 项目设计（自上而下）

### 小程序功能设计

1. 多人对多设备的添加和访问。
   1. 每个人登陆到小程序后都可以绑定自己的手机号，然后通过扫码或者其他人分享的的方式绑定一些设备。
   2. 对于绑定的所有的设备，该用户都有对设备的访问权限，可查看的信息有：
      1. 设备开度（0-100%）
      2. 远控开关（打开/关闭)
      3. 控制精度（0-100）
      4. 运行状态
      5. 网络状态
      6. 故障信息
      7. 设备名和设备备注信息
      8. 设备序列号
      9. 设备日志
2. 多人对多设备的控制能力。
   1. 对于对设备有控制权限的用户，他们可以修改设备的一些参数和配置信息，并且可以生成相应的日志。
   2. 多人对同一设备的修改会互相同步，并且如果权限被取消了也会在修改前进行判断，以此保证逻辑的准确性。
   3. 有控制权限的用户可以修改的参数有：
      1. 设备开度（0-100%）
      2. 远控开关（打开/关闭)
      3. 控制精度（0-100）
      4. 设备名和设备备注信息
3. 管理员的特殊权限。
   1. 绑定设备的第一个人成为该设备的管理员，拥有该设备的所有权限。
   2. 管理员可以开启设备的访问控制开关。
      1. 默认设备初始状态都未开启访问控制开关的，未开启访问控制开关的设备可以被所有绑定的用户修改。
      2. 如果管理员开启访问控制权限，所有用户只有通过管理员分享控制权限或者输入控制密码才能再次获得对该设备的控制权限。
   3. 管理员可以管理修改设备的控制密码。
      1. 其他只拥有访问权限的用户可以通过控制密码来或者设备的控制权限。
      2. 但该控制权限不可以分享。
      3. 如果管理员修改了设备的控制密码，那么其他通过密码获得控制权限的用户将会失去控制权限。
4. 多人对绑定设备的共享能力。
   1. 所有人都可以对设备进行分享。
   2. 但只有管理员可以分享控制权限，其他人只可以通过手机号分享访问权限。
   3. 被分享的用户可以继续将访问权限分享给其他人。
5. 设备的权限管理能力。
   1. 管理员拥有对该设备的权限管理能力。
   2. 在权限管理页面，管理员可以看到该设备所有绑定的用户，也可以看到权限的来源以及是否授权，授权人是谁。
   3. 管理员可以选择任意用户的权限进行删除，删除之后，该用户将失去对该设备的相关权限。

### 小程序页面设计

1. 设备首页
   1. 头像+手机号+设置(微信号，手机号（修改）)
   2. 设备列表
      1. 设备名/备注名
      2. 阀门开度
      3. 运行状态
      4. 是否可控制
      5. 是否管理员
   3. 添加设备
      1. 扫描添加
2. 设备界面(请求设备信息 userDevice)
   1. 设备信息
      1. 阀门开度(0-100)
      2. 开关量控制(0-1)
      3. 调节精度(0-1000)
      4. 阀门状态
      5. 网络连接
      6. 故障信息
   2. 设备配置
      1. 管理员账户
      2. 名称(路人不可编辑)
      3. 备注名称(可编辑)
      4. 序列号
      5. 备注信息(路人不可编辑)
      6. 管理员可见 - 设备共享 (获取sense场景值)
      7. 管理员可见 - 启用访问控制权限(普通用户不显示)
      8. 管理员可见 - 控制密码(输入不一样才修改)
      9. 管理员可见 - 用户权限管理
      10. 删除设备
      11. 权限来源 （0 代表扫码，1代表输入序列号，2被授权）
      12. 授权权限 （0 代表仅访问， 1 代表访问+控制，默认0）
      13. 密码权限 （0 代表无密码，仅访问，1代表有密码，访问—+控制，默认为0）
      14. 授权者账户
   3. 设备日志
      1. 日期
      2. 时间
      3. 操作人和操作内容
3. 个人设置
   1. 微信头像
   2. 微信昵称
   3. 服务器ip配置
   4. 手机号绑定

### 小程序交互逻辑的设计

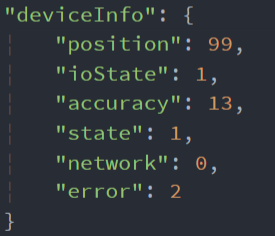
1. 登陆逻辑
   1. 那两张图
   2. 进入之前请求openid，等待用户确认，然后绑定手机号。以后存储该openid
2. 首页
   1. 登录时请求设备列表
   2. 下拉时刷新设备列表
3. 添加设备图标
   1. 点击启用手机摄像头
   2. 扫码成功添加设备
   3. 刷新首页设备列表
4. 点击设备
   1. 请求设备信息和设备配置数据
5. 设备页面
   1. 下拉更新当前界面信息
   2. 点击设备日志页面请求设备日志
   3. 点击可修改项提示输入框
      1. 点击确定判断权限
      2. 点击确认向发送修改的GET请求
      3. 点击取消撤销操作
      4. 修改成功弹出一秒的提示，并且更新当前界面
      5. 权限失效修改失败弹出三秒的提示
   4. 管理员点击启用访问控制权限的开关
      1. 向服务器发送启用的GET请求
   5. 管理员修改设备的控制密码
      1. 向服务器发送修改密码的GET请求
      2. 同时更新当前显示的密码
6. 权限管理页面
   1. 选择要删除的权限拥有者
   2. 删除成功后会更新当前界面
7. 分享界面
   1. 有仅访问和访问和控制两种互斥的开关
   2. 选择一个后可以输入手机号进行分享
   3. 发送增加权限的GET请求
8. 个人设置页面
   1. 可以修改服务器的ip地址
   2. 服务器连接异常有警告标识
   3. 手机号未绑定也有警告标识
   4. 输入手机号或ip地址发送绑定请求

### 小程序数据结构设计

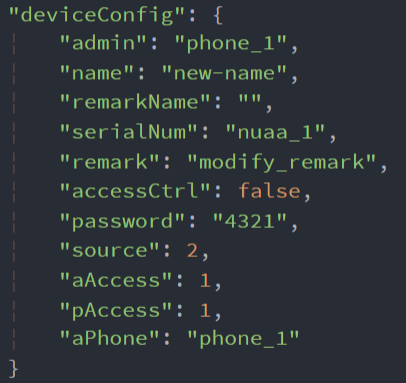
1. deviceRight设备简要信息



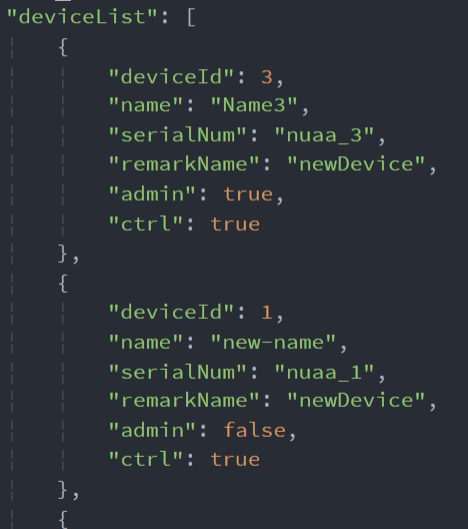
1. DeviceInfo设备运行参数



1. DeviceConfig设备配置参数



1. DeviceList用户设备列表



1. RightList设备的用户权限列表



1. DeviceLog设备日志

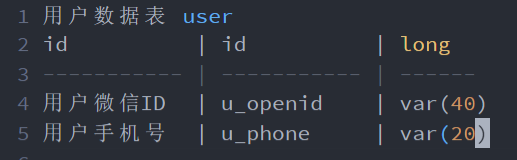
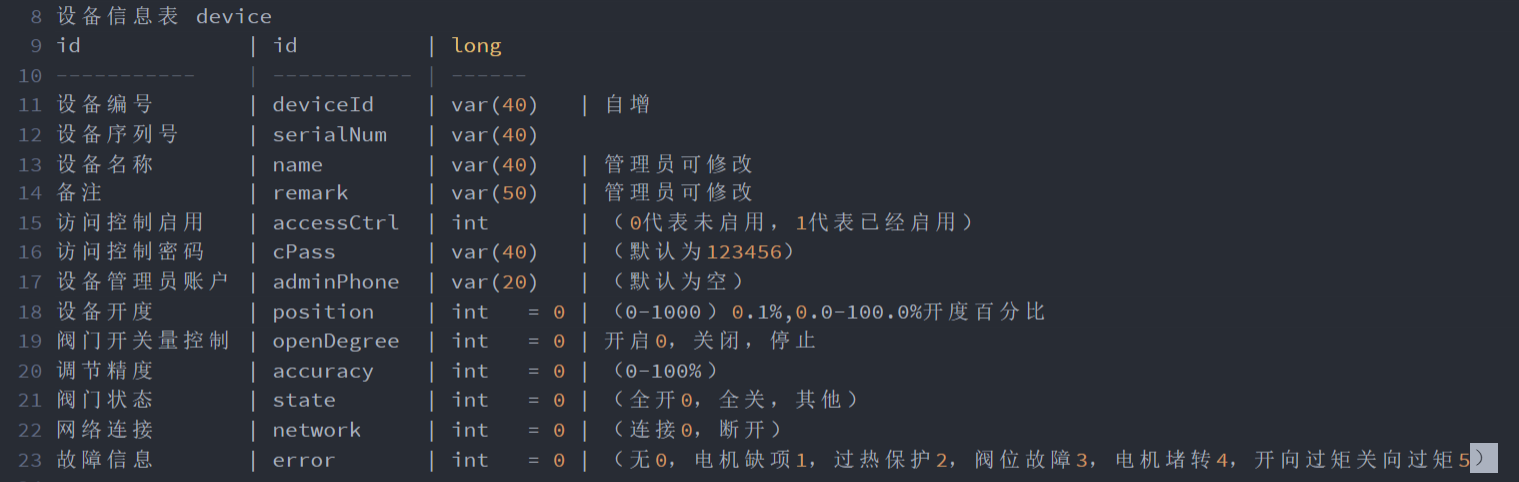
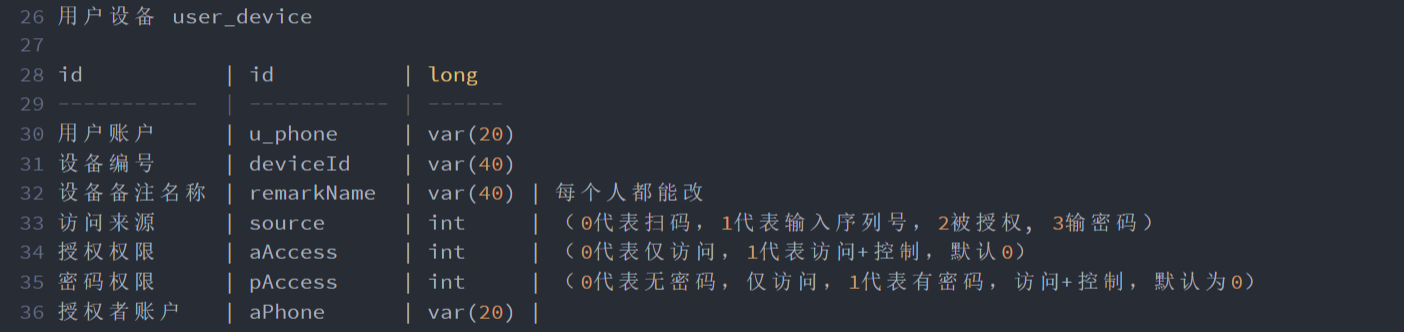


### 服务器通信接口设计



### 数据库设计

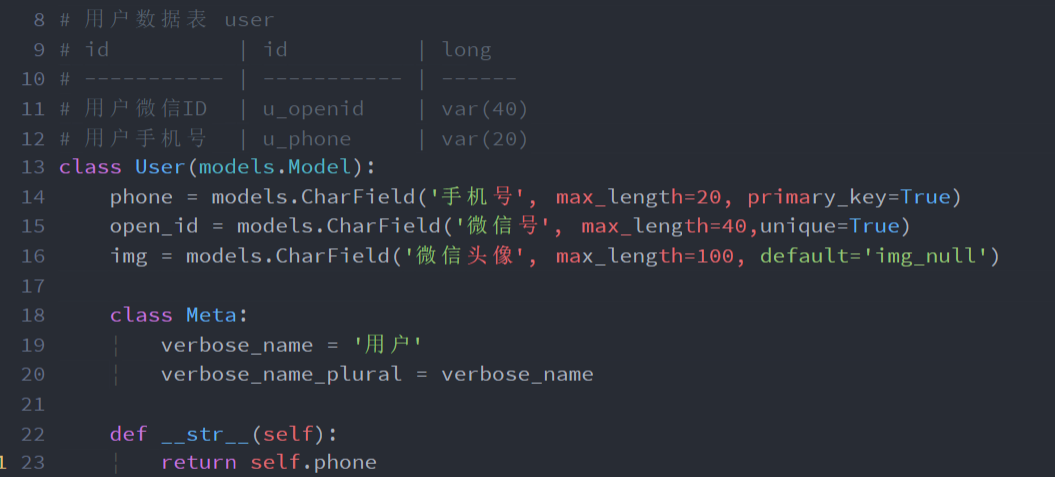
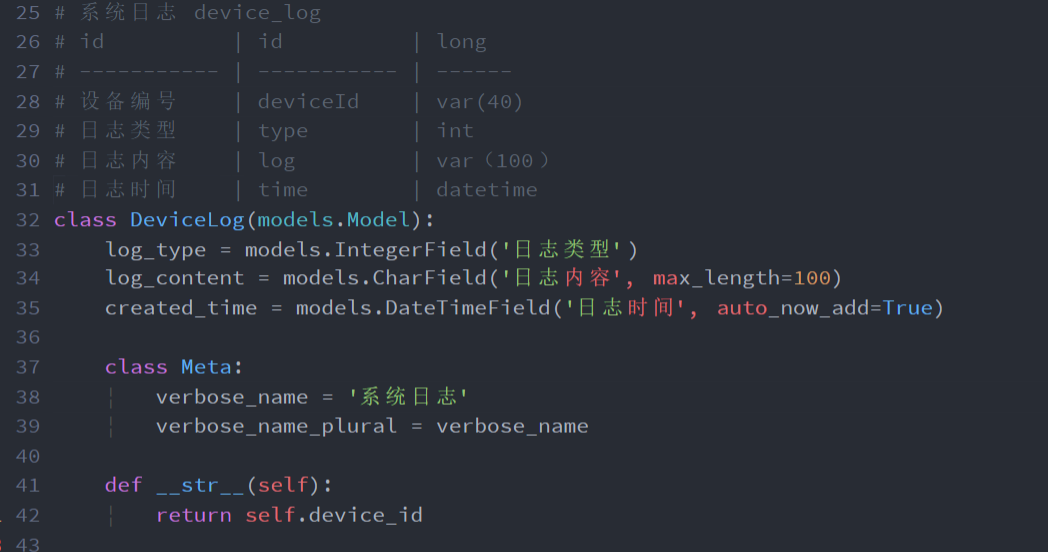
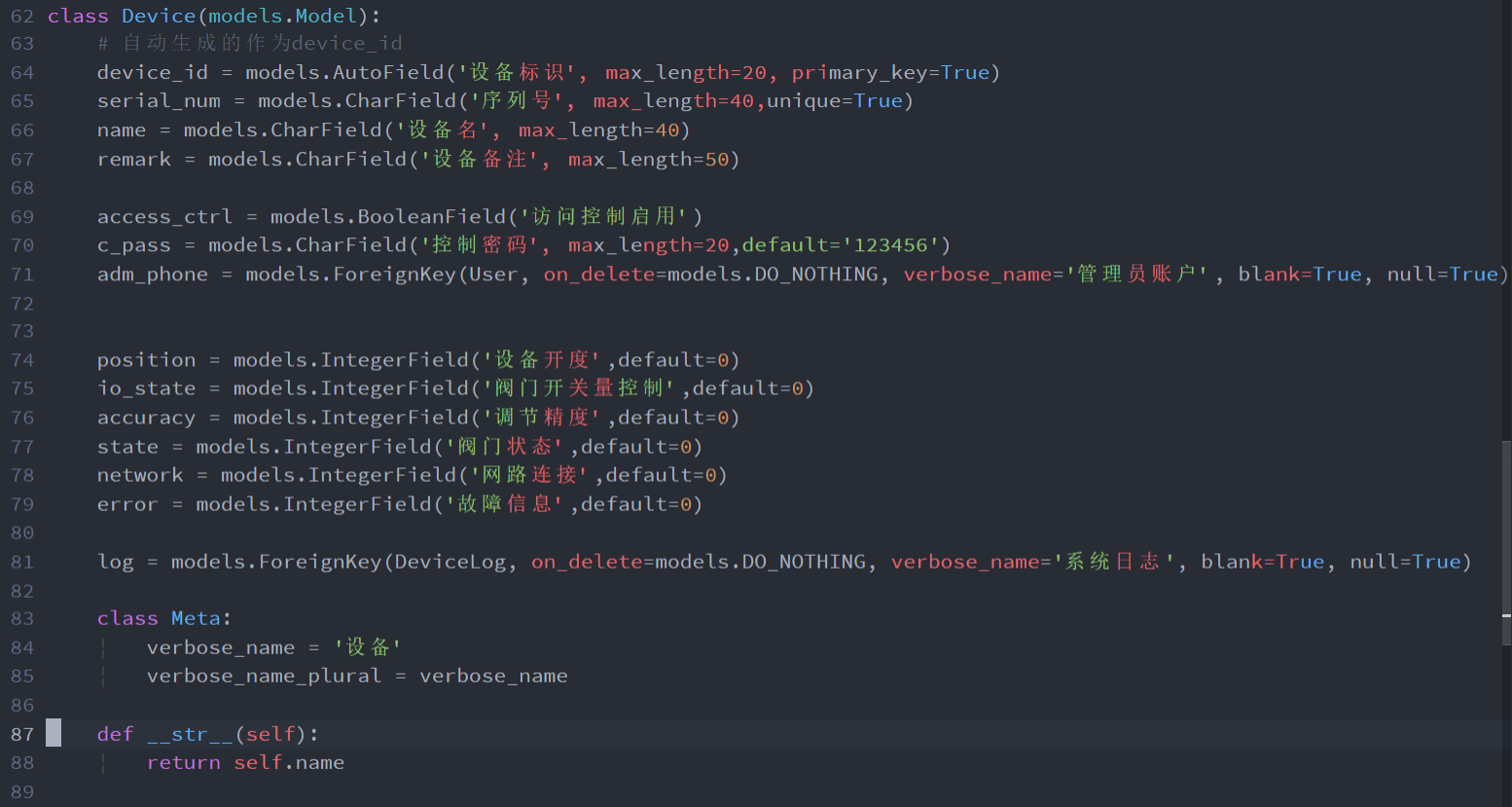
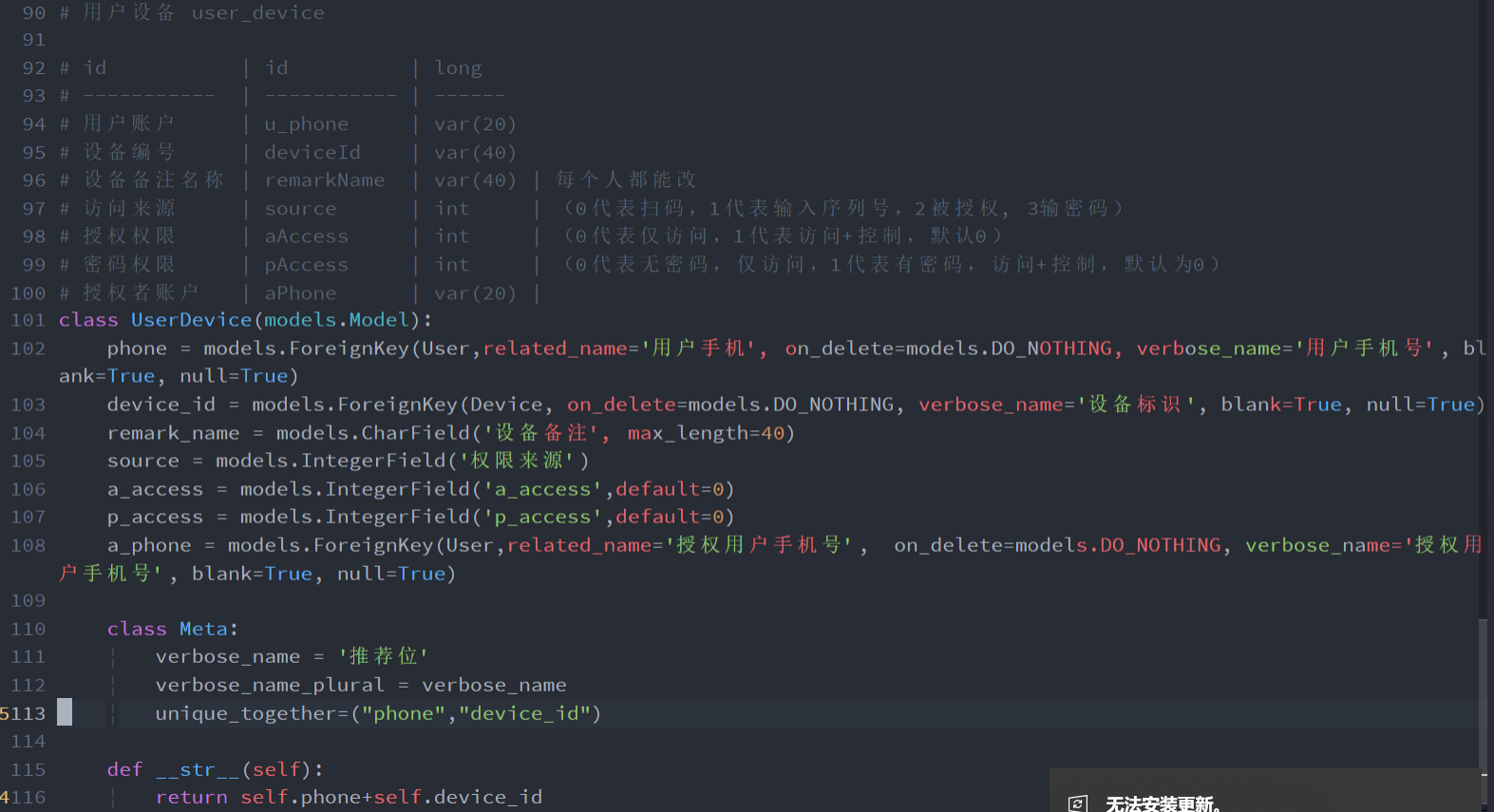
根据上述的需求所涉及的数据和数据之间的对应关系，我们建立了如下几个数据表。

1. 用户数据表
   1. 
2. 设备配置表
   1. 
3. 用户设备表
   1. 
4. 设备日志表
   1. 

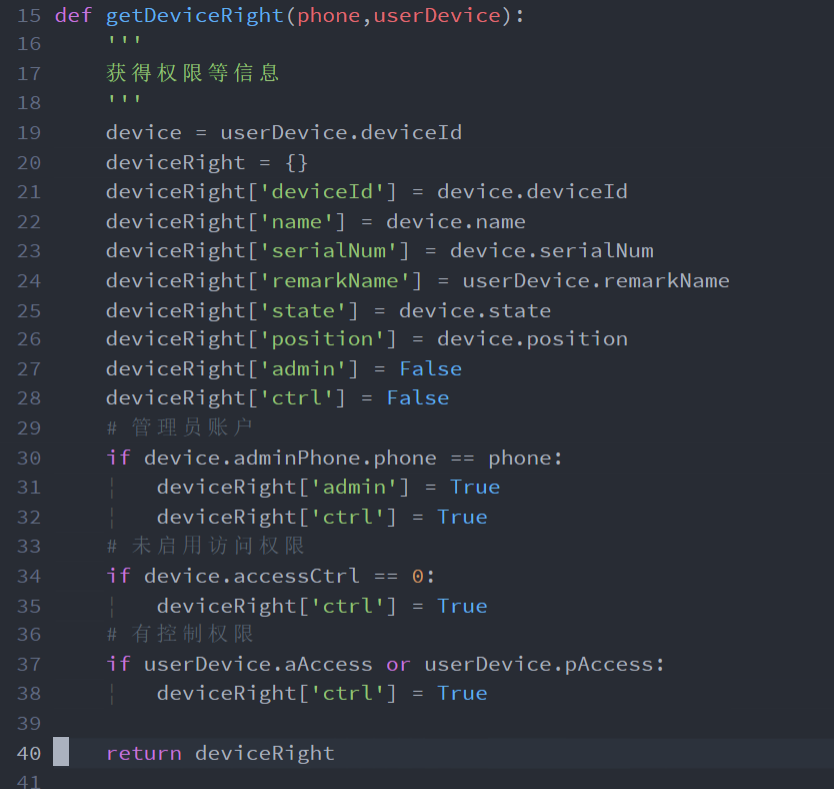
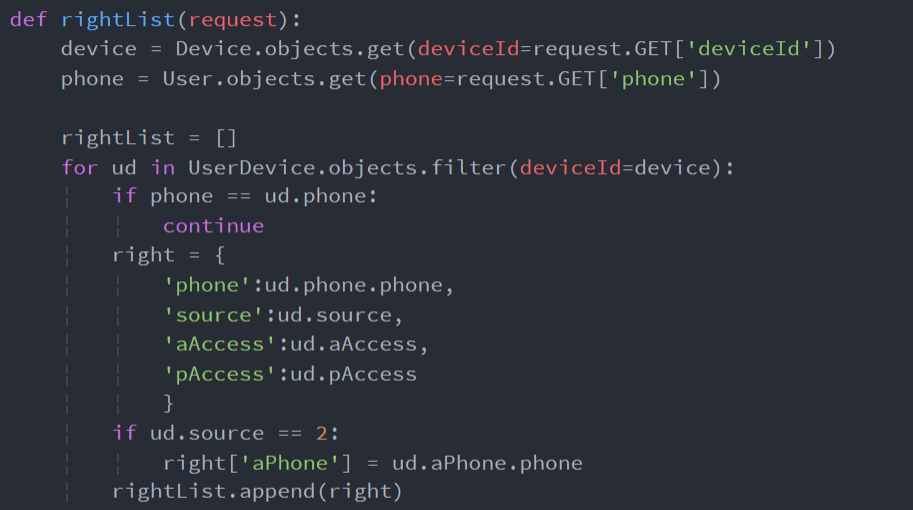
## 项目开发（自下而上）

由于django框架（优点），我们选择django框架来开发我们的后台，而不是云开发模式

### 数据库模型

1. User用户表
   1. 
2. DeviceLog设备日志表
   1. 
3. Device设备表
   1. 
4. 用户设备表UserDevice
   1. 

### 小程序所需数据结构的获取方法

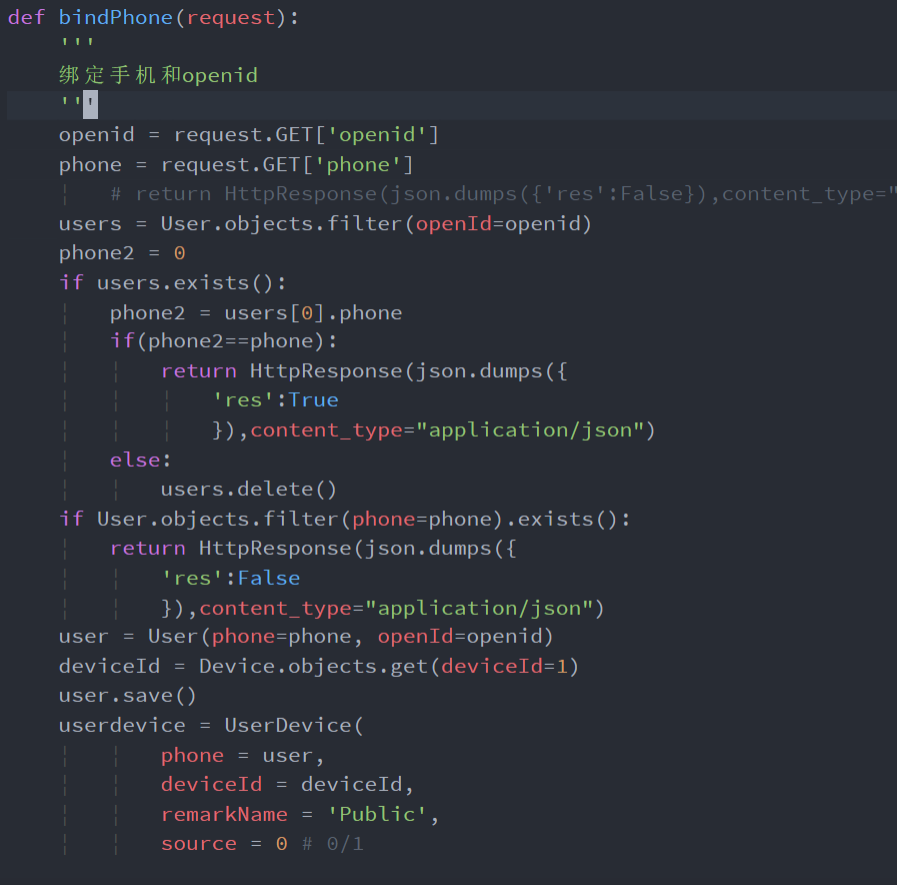
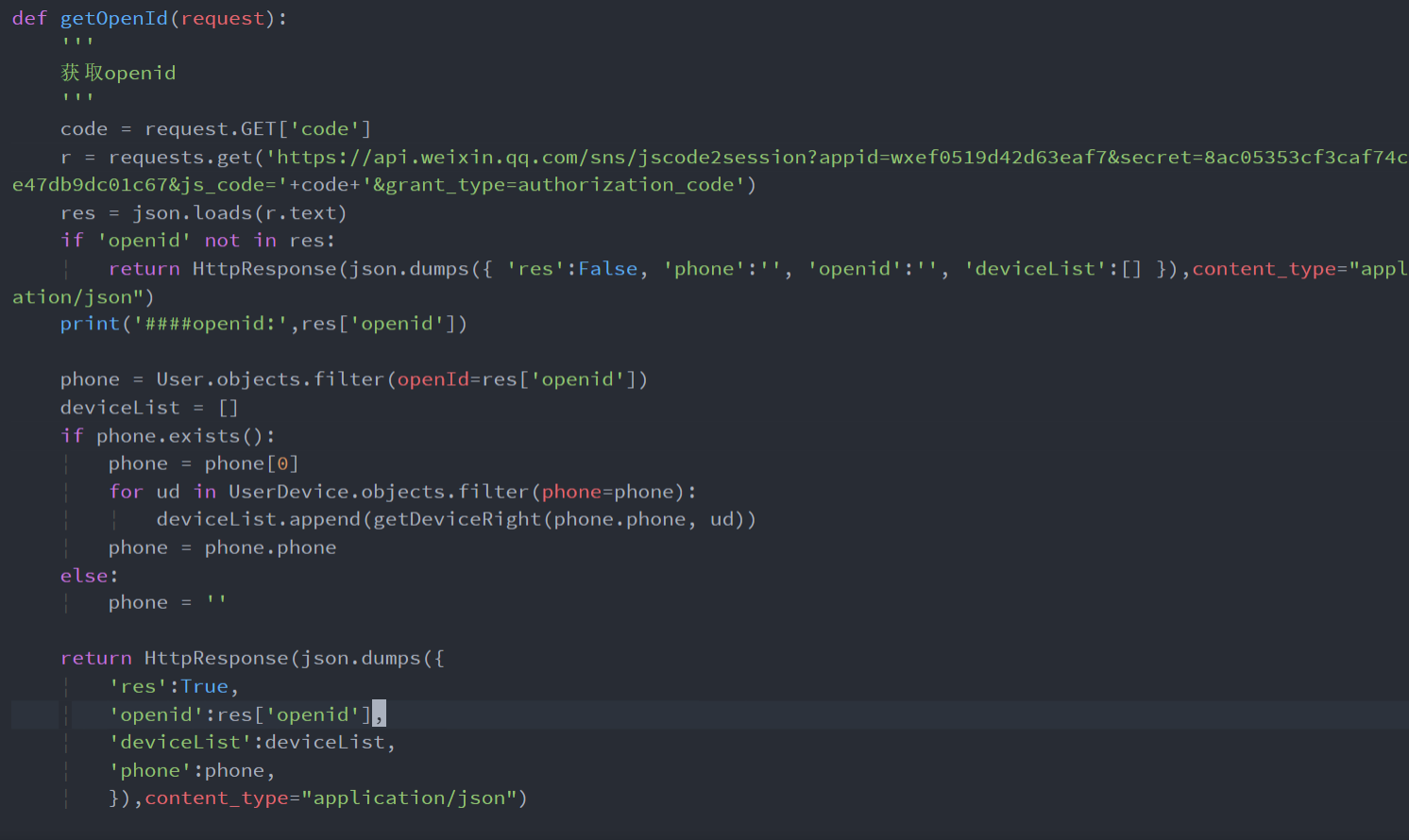
1. 获取设备运行参数页面的数据结构
   1. 
2. 获取设备配置参数界面的数据结构
   1. 
3. 获取首页设备列表中设备简要信息的数据结构
   1. 
4. 获取设备所有绑定用户权限详情的数据结构
   1. 
5. 获取设备日志的数据结构
   1. 

### Web接口实现与测试

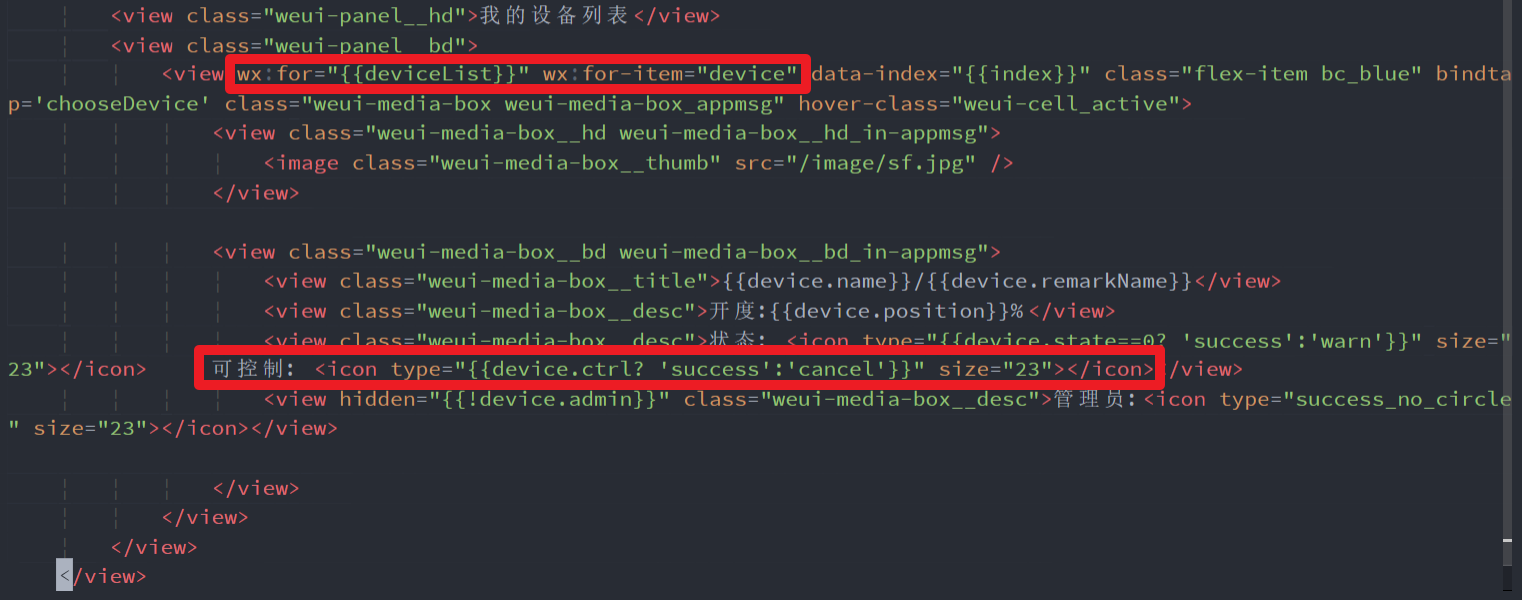
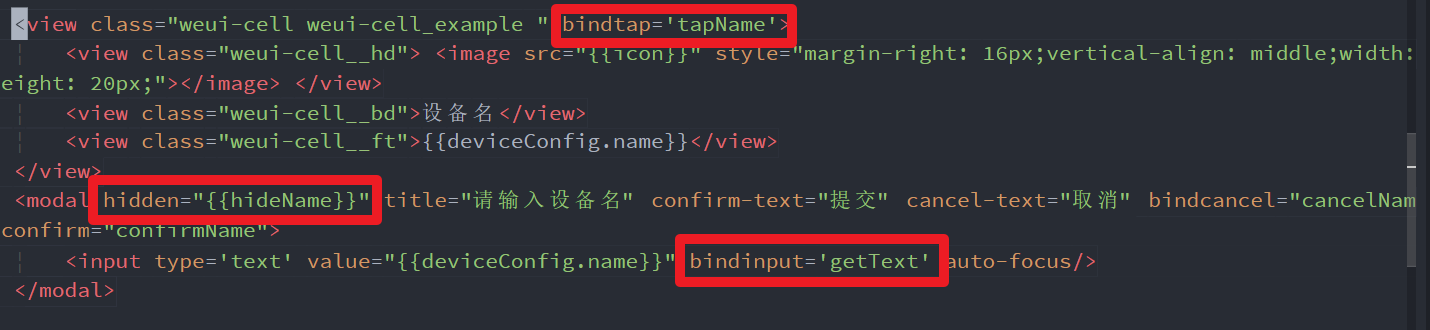
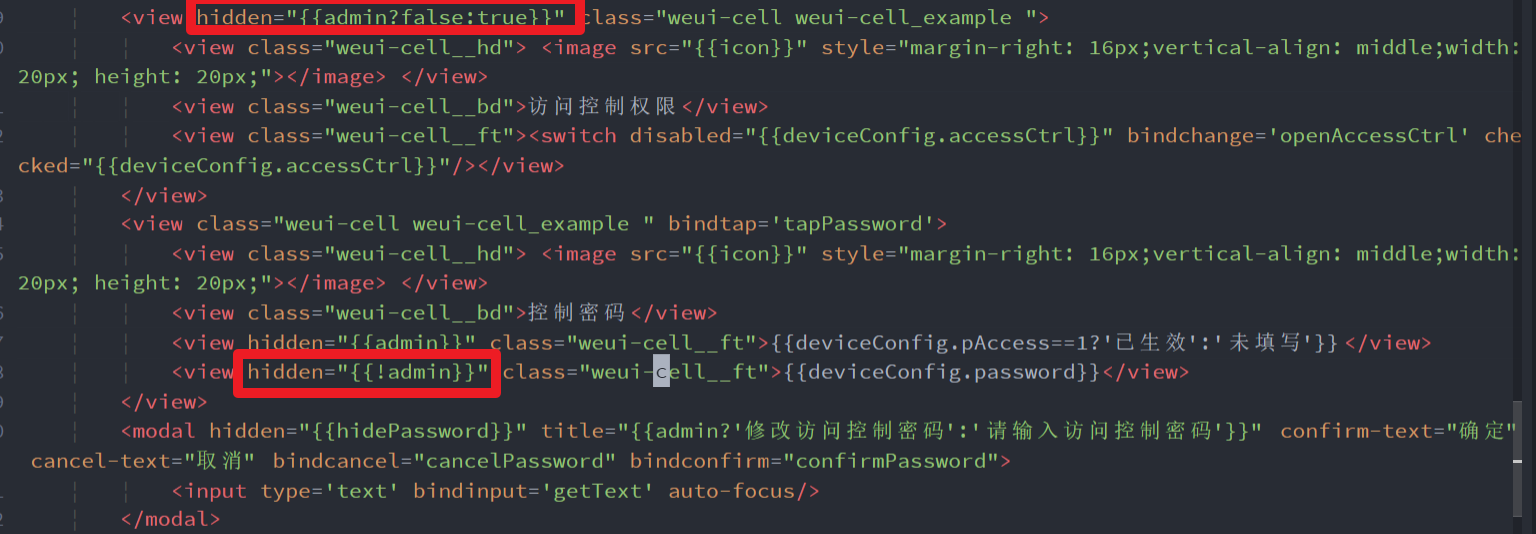
路由截图：



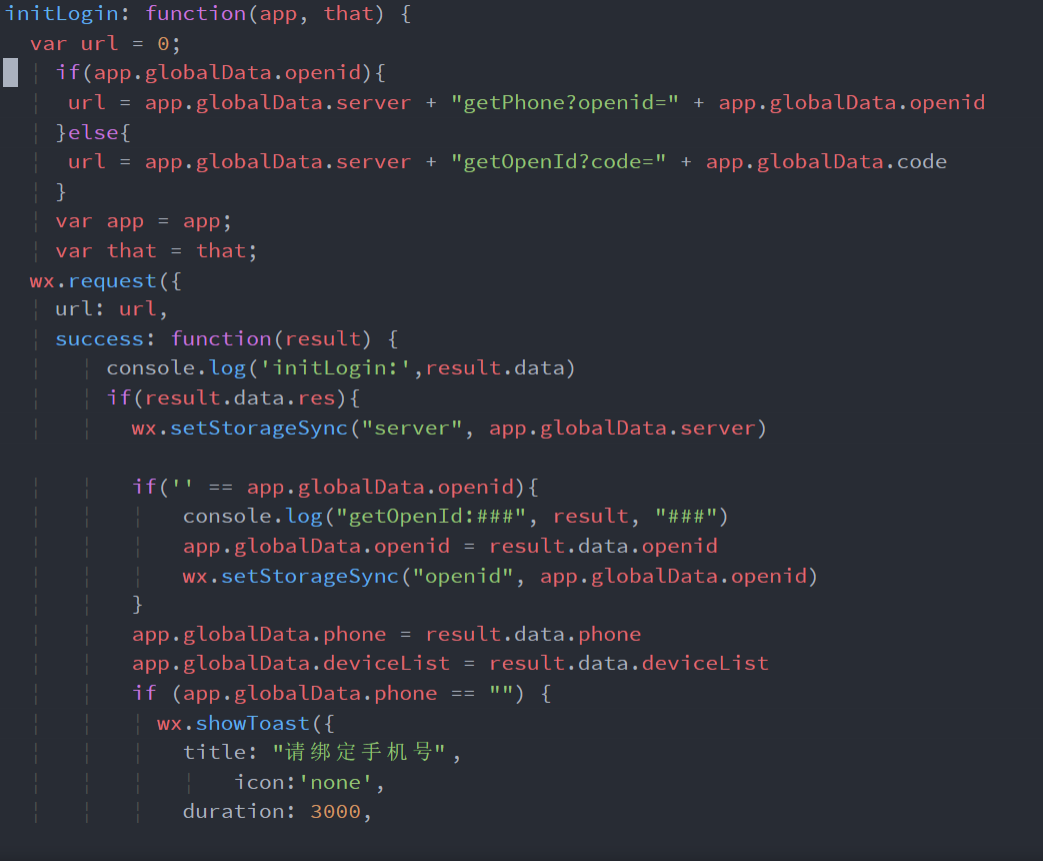
接口详情：

1. 获取用户的设备列表
   1. 
2. 绑定手机号
   1. 
3. 获取微信openid和手机号
   1. 
4. 增加设备
   1. 
5. 删除设备
   1. 略
6. 设备运行参数
7. 设备配置参数
8. 修改设备相关参数的处理接口
9. 增加访问权限
10. 增加控制权限
11. 启用访问控制
12. 修改设备密码
13. 删除用户的设备权限
14. 获得权限列表

### 小程序交互逻辑实现

1. 设备列表循环显示
   1. 
2. 设备参数修改弹出输入窗口
   1. 
3. 权限管理页面多选框
   1. 
4. 通过模块隐藏属性实现管理和普通用户的不同页面
   1. 

### 小程序接口实现

1. 登录
   1. 
   2. 
2. 首页
   1. 登录时请求设备列表
      1. 
3. 点击设备
   1. 请求设备信息和设备配置数据
      1. 
      2. 
4. 设备页面
   1. 点击设备日志页面请求设备日志
      1. 
   2. 点击可修改项提示输入框
      1. 点击确定判断权限
         1. 
      2. 点击确认向发送修改的GET请求
         1. 
      3. 修改成功弹出一秒的提示，并且更新当前界面
      4. 权限失效修改失败弹出三秒的提示
         1. 